JAN/FEB 1988 ISSUE # 16

Hunter Valley

VZ

Journal



PRODUCED BI-MONTHLY BY H.V.VZ.U.G.

The program is just a simple one. Words are held in DATA statements at end of listing. Any family member can read out words for student to spell. When changing words make sure you have 20, otherwise you'll have to reDIMension.

Fed up with the tinny sound from your VZ, then change it for one with improved sound and volume. Even a beginner should be able to do this project.

This is the basic version of FIND routine which is a very useful utility to have around. It can be even better if it's available on power up. With Larry's help it now lives in my 2K ZERO POWER RAM at 6000H-67FFH, Thanks Larry.

Last issue featured M/C version while this one is in basic. A simple demonstration program is included to show what could be achieved. I'm sure you'll be able to adapt it to your own needs.

As I keep club's membership list on DISKLIST I keep trying to improve it's operation. The two main improvements are to SAVE/LOAD toutines. You can now update existing file without using the ERAse function first. Also file MERGING has been incorporated as well.

A pointer change routine has been added to M/WORDS program so words like TRON, TROFF, etc can be activated. Ideas on how to put routines in 2K BIB RAM are presented as well. Don't forget, the main purpose of the BIB RAMS is for software developement. My thanks to Dave Mitchell and Robert Quinn and Larry Taylor for their help with above.

Turn your PRINTER/PLOTTER into a typewriter printing UPPER/lower case, INVERSE, GRAPHICS and screen dumps as well.

If some of your keys don't work too well then don't rush out and buy a M/Keyboard before trying out Dick Smith's magic spray which just might do the trick and is much cheaper.

Although the second block move routine presented, it's use would be better suited to programs like M/WORDS, EPROM PROGRAMMERS, etc. Chose one or both according to your applications.

Matthew, our resident GRAPHICS expert shows of his skills. He also designed an exellent Graphics Editor which is simple to use and full of features. Both tape and disk users are catered for.

BELIEVE IT OR NOT :-

A person new to computers received a VZ300 for Christmas. The persons delight turned to dismay upon spotting the MADE IN HONG KONG label on the bottom. The person took it back to exchange for one made locally, the reason, person couldn't read chinese.

PS :- Names and places have been changed to protect the innocent.

I hope all our readers had a happy and safe festive season and a good start to 1988 and here's hoping it will stay that way.

FOR PRIVATE SALE 1 :-

1 OFF VZ 300 - 1 OFF VZ 200 DATASSETTE - 1 OFF VZ 200 JOYSTICKS.
Includes leads and powerpacks. All used and in good order.
Price as UNIT - $120.00 - Includes POSTAGE.
Contact Brian GREEVE 37 NORWOOD ROAD RIVERVALE 6103
PERTH WESTERN AUSTRALIA.

FOR PRIVATE SALE 2 :-

1 OFF LIGHT (GRAPHICS) PEN - As new and still in original packing. $32.00
Includes POSTAGE - Ring Dave BOYCE on (08) 384 6574.

CLUB DEMONSTRATION :-
At the March club meeting Joe LEON will demonstrate the end result of his BIB RAMS articles by transfering information to EPROM and installing in his VZ 300 for a working demonstration.

VZ USER GROUPS AND PUBLICATIONS

VZ USER MARK HARWOOD P.O.BOX 154 DURAL N.S.W. 2158

LE'VZ OOP J.C.E. D'ALTON 39 AGNES St. TOOWONG QLND. 4066
VSOFTWAREZ - SOFTWARE/HARDWARE FOR SALE

VZ DOWN UNDER SCOTT LE BRUN 59 BRENTWOOD DVE WANTIRNA 3152
SOFTWARE/HARDWARE FOR SALE

VZ-LINK - PETER J. HILL P.O.BOX 1972 C.P.O. AUCKLAND N.Z.

WAVZ - GRAEME BYWATER P.O. BOX 388, MORLEY W.A. 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178

HUNTER VALLEY VZ USERS' GROUP - P.O. BOX 161 JESMOND N.S.W. 2299
EDITOR-JOE LEON (049)51 2756 - SECRETARY-ROSS WOODS (049)71 2843

SUBSCRIPTION - H.V.VZ.JOURNAL - 6 MONTHS $9.00 - 12 MONTHS $18.00
New Zealand - 6 MONTHS $12.00 - 12 MONTHS $24.00

NEW VENUE - NEW DATES - NEW VENUE - NEW DATES - NEW VENUE

MEETINGS - FIRST FRIDAY of MONTH at JESMOND NEIGHBOURHOOD CENTRE
MORDUE PARADE - REAR STOCKLAND MALL (BIG W) JESMOND

NOTE :- When writing to any above or H.V.VZ. Users' Group for information please enclose a S.S.A.E. or 2 Int. Reply Coupons.

```
10 '*************************************************************
15 '*    TRAINING PROGRAM FOR LEARNING TO SPELL NEW WORDS     *
20 '*  DESIGNED FOR PARENT PARTICIPATION -- BY P AND J LEON  *
25 '*************************************************************
30 :
35 '*************************************************************
40 '* THIS PROGRAM IS FAIRLY EASY TO USE AS IT'S MENU DRIVEN *
45 '* AFTER RUNNING PROGRAM SELECT NUMBER OF WORDS TO SPELL. *
50 '* NEXT EITHER A PARENT, BROTHER, SISTER CAN READ OUT THE *
55 '* WORDS THE STUDENT WISHES TO LEARN HOW TO SPELL. THE VZ *
60 '* CHECKS THE SPELLING AUTOMATICALLY INFORMING STUDENT    *
65 '* WHETHER HE/SHE IS RIGHT/WRONG AND PERCENTAGE CORRECT.  *
70 '*************************************************************
75 :
80 '*************************************************************
85 '* LINES 800 TO 830 CONTAIN THE 20 WORDS STUDENT WANTS TO *
90 '* LEARN.  JUST REPLACE WITH 20 NEW WORDS OF YOUR CHOICE. *
95 '*************************************************************
100 :
105 POKE30744,96:COLOR,0:DIM E$(20):GOSUB150:GOSUB700
110 FORI=1TO20:READE$(I):NEXT I:GOTO200
140 :
150 CLS:COLOR7:PRINT TAB(6)" SPELLING TRAINER "
155 PRINT" =============================="
160 PRINT"  RIGHT >       WRONG >":PRINT
165 PRINT"  PERCENTAGE CORRECT ->"
170 PRINT" =============================="
175 PRINT@77,"0":PRINT@93,"0":PRINT@154,"0.00":RETURN
190 :
200 PRINT@226,"QUESTION  #     WAS":PRINT@418,"QUESTION #"
210 FORI=1TONW:PRINT@429,USING" ##";I:GOSUB600
220 PRINT@482," PRESS 'Q' TO QUIT ";
290 :
300 A$="":PRINT@322,;:INPUT" SPELLING  ";A$:IFA$=""THEN 300
310 IFA$="Q"THENCLS:END
320 C=0:FORJ=1TO20:IFE$(J)=A$THENC=1:NEXTJELSENEXTJ
330 IFC=1THENGOSUB500ELSEGOSUB510
340 NEXTI:GOSUB600
390 :
400 PRINT@443,"Y      ";
410 PRINT@418,;:INPUT" LIKE TO TRY AGAIN    ";T$
420 IF T$="Y"THENRUNELSEIFT$="Q"THENCLS:ENDELSE400
490 :
500 PRINT@247," RIGHT ":SOUND31,1:R=R+1:GOTO520
510 PRINT@247," WRONG ":SOUND15,4:W=W+1
520 PRINT@76,USING"##";R:PRINT@92,USING"##";W
530 PRINT@152,USING"###.##";R/I*100
540 PRINT@237,USING" ##";I:RETURN
590 :
600 FOR J=288TO388STEP32
610 PRINT@J,"                                ";:REM 32 SPACES
620 NEXT:RETURN
690 :
700 GOSUB600:PRINT@289," NUMBER OF WORDS = 5 TO 20  "
710 PRINT@380,"5"
720 PRINT@351,;:INPUT"   SELECT NUMBER OF WORDS  ";NW
730 IFNW <5 ORNW >20 THEN700ELSERETURN
790 :
800 DATA ACCOMPLISHED,BEHAVIOUR,COMMONWEALTH,DAYLIGHT,ELEPHANT
810 DATA FINANCIAL,GENERATION,HEADPHONES,IMITATE,JUSTIFIED
820 DATA KNIFE,LENGTH,MASTERPIECE,NUISANCE,OCCUPATION
830 DATA PHARAOH,QUIET,RECEIPT,SARCASM,WHISTLE
```

IMPROVING VZ 200/300 SOUND QUALITY :-

The reason for the tinny sound in the VZ is the PIEZO TRANSDUCER. It's located on the bottom of the case under the P.C.B., and because it's glued down it further reduces the sound output. This modification can be done by the beginner.

Before starting the mod you'll have to purchase one of the two sound transducers or 2"" speaker described below and or a miniature pot.

1) 5.1cm (2") 8 Ohm speaker. TANDY Cat. No. 40-245 - $3.99
2) PIEZO TRANSDUCER 1. - 30mm X 5mm - Available from D.Smith.
3) PIEZO TRANSDUCER 2 - 30mm X 10mm - TANDY No. 273-073 - $2.99
4) MINIATURE ROTARY POT (5K) - Available from D.SMITH - $1.20

I tried all three and found the 30mm X 10mm P/TRANSDUCER the most satisfactory as far as sound output, size and price. Sound output between the 5.1cm speaker and 30mm X 10mm P/TRANSDUCER was nearly identical. So size and price took priority.

The P/TRANSDUCER element used in the VZ is the same as the two described previously. The one in the VZ is uncased, while the other two are cased. the larger the case the more sound output. And now to the mod.

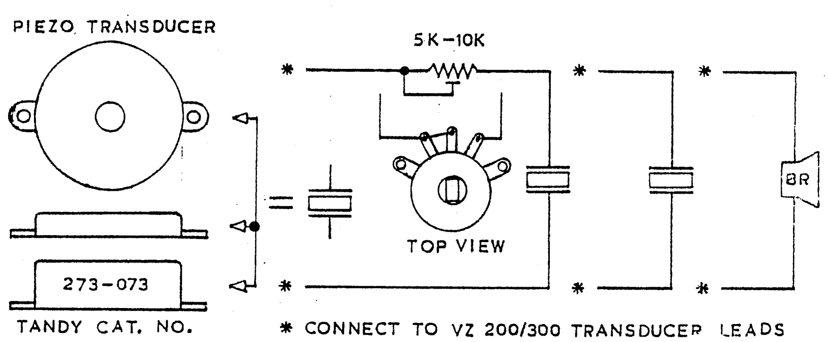
Turn your VZ upside down and remove all the screws holding the two halves of the case together and turn VZ right way up and back to front. Lift top of case and put towards the back.

The screws holding the P.C.B. in place will have to be removed next. Lift the P.C.B. out carefully and you should see the P/TRANSDUCER element glued to bottom of case. Unsolder or cut from the element the two leads going to it.

If you desire a volume control, now is the time to insert it in one of the leads as shown in circuit diagram. It can be mounted on bottom left lip of the case. Drill a hole to suit the shaft and mount it upside down with the knob going under the lip. The volume control is linear in operation and at mimimum is like the volume from original VZ while at Maximum is fair bit louder.

Next solder the leads from your new TRANSDUCER to the ends of the unsoldered leads. If you have'nt a soldering iron then simply twist the wires together. Make sure you insulate the joined wires from each other and the VZ by using sleeving or wrapping some electrical tape around the joints.

The best place to put the P/TRANSDUCER is at front left of case with the leads going under the P.C.B. Use a bit of tape on one side of P/TRANSDUCER to keep it in place. All that's left now is to reasemble. I think you'll agree, it was'nt hard.



```
100 REM #######################################################
105 REM # MODIFIED FIND UTILITY BY LARRY TAYLOR 1985          #
110 REM # ADAPTED FROM ARTICLE BY C. STAMBOULIDIS, PUBLISHED  #
115 REM # IN PERSONAL COMPUTER GAMES IN APRIL, 1985.          #
120 REM #######################################################
125 REM
130 REM #######################################################
135 REM # THE ROUTINE IS USED TO LOCATE A STRING OF CHARACTERS#
140 REM # WITHIN A BASIC PROGRAM. THE ROUTINE IS CALLED WITH  #
145 REM # PRINT& FOLLOWED BY THE STRING IN INVERTED COMMAS.   #
150 REM # ANY LEADING SPACES IN THE STRING WILL BE IGNORED.   #
155 REM # THE NUMBER OF THE LINE, CONTAINING THE STRING AND  #
160 REM # ITS LOCATION IN MEMORY, WILL BE PRINTED ON SCREEN. #
165 REM #######################################################
170 REM
175 REM #######################################################
180 REM # CALCULATE THE NEW TOP OF MEMORY POINTER             #
185 REM #######################################################
187 REM
190 NB=133:TM=(PEEK(30897)+PEEK(30898)*256)-NB
195 HB=INT(TM/256):LB=TM-HB*256
200 POKE30897,LB:POKE30898,HB
202 REM
205 REM #######################################################
210 REM # LOAD '&' VECTOR WITH TOP OF MEMORY PLUS ONE         #
215 REM #######################################################
217 REM
220 POKE31125,LB+1:POKE31126,HB
222 REM
225 REM #######################################################
230 REM # RESET THE BASIC STACK POINTER                       #
235 REM #######################################################
237 REM
240 CLEAR50
242 REM
245 REM #######################################################
250 REM # GET NEW TOP OF MEMORY AND MOVE TO NEXT LOCATION     #
255 REM #######################################################
257 REM
260 TM=PEEK(30897)+PEEK(30898)*256
265 IFTM>32767THENTM=TM-65536
267 REM
270 REM #######################################################
275 REM # LOAD 133 BYTE FIND ROUTINE                          #
280 REM #######################################################
282 REM
285 FOR T=1TO133
290  READD
295  POKETM+T,D
300  CS=CS+D
305 NEXT
310 REM #######################################################
315 REM # IF DATA CHECKSUM VERIFIES EXECUTE SET UP PROGRAM    #
320 REM #######################################################
322 REM
325 IFCS<>13857THENPRINT"- ERROR IN DATA ENTRY -":END
330 POKE 31124,195
332 REM
335 REM #######################################################
340 REM # FIND ROUTINE DATA IN DECIMAL FORM                   #
345 REM #######################################################
```

```
350 DATA 35              :' INC HL
355 DATA 205,140,53      :' CALL 358CH
360 DATA 58,214,122      :' LD A,(7AD6H)
365 DATA 61              :' DEC A
370 DATA 50,214,122      :' LD (7AD6H),A
375 DATA 183             :' OR A
380 DATA 40,116          :' JR Z,74H
385 DATA 221,42,164,120  :' LD IX,(78A4H)
390 DATA 221,126,0       :' LD A,(IX+00H)
395 DATA 183             :' OR A
400 DATA 32,6            :' JR NZ,06H
405 DATA 221,126,1       :' LD A,(IX+01H)
410 DATA 183.            :' OR A
415 DATA 40,100          :' JR Z,64H
420 DATA 221,110,0       :' LD L,(IX+00H)
425 DATA 221,102,1       :' LD H,(IX+01H)
430 DATA 34,176,121      :' LD (79B0H),HL
435 DATA 221,110,2       :' LD L,(IX+02H)
440 DATA 221,102,3       :' LD H,(IX+03H)
445 DATA 34,173,121      :' LD (79ADH),HL
450 DATA 221,229         :' PUSH IX
455 DATA 225             :' POP HL
460 DATA 35              :' INC HL
465 DATA 35              :' INC HL
470 DATA 35              :' INC HL
475 DATA 35              :' INC HL
480 DATA 34,40,122       :' LD (7A28H),HL
485 DATA 205,126,43      :' CALL 2B7EH
490 DATA 17,232,121      :' LD DE,79E8H
495 DATA 58,214,122      :' LD A,(7AD6H)
500 DATA 71              :' LD B,A
505 DATA 33,156,122      :' LD HL,7A9CH
510 DATA 35              :' INC HL
515 DATA 126             :' LD A,(HL)
520 DATA 183             :' OR A
525 DATA 40,54           :' JR Z,36H
530 DATA 26              :' LD A,(DE)
535 DATA 183             :' OR A
540 DATA 40,37           :' JR Z,25H
545 DATA 19              :' INC DE
550 DATA 190             :' CP (HL)
555 DATA 32,236          :' JR NZ,ECH
560 DATA 16,241          :' DJNZ F1H
565 DATA 62,32           :' LD A,20H
570 DATA 205,58,3        :' CALL 033AH
575 DATA 42,173,121      :' LD HL,(79ADH)
580 DATA 205,175,15      :' CALL 0FAFH
585 DATA 62,32           :' LD A,20H
590 DATA 205,58,3        :' CALL 033AH
595 DATA 42,40,122       :' LD HL,(7A28H)
600 DATA 205,175,15      :' CALL 0FAFH
605 DATA 58,174,122      :' LD A,(7AAEH)
610 DATA 192,11          :' ADD A,0BH
615 DATA 230,32          :' AND 20H
620 DATA 32,6            :' JR NZ,06H
625 DATA 221,42,176,121  :' LD IX,(79B0H)
630 DATA 24,151          :' JR 97H
635 DATA 62,13           :' LD A,0DH
640 DATA 205,58,3        :' CALL 033AH
645 DATA 24,243          :' JR F3H
650 DATA 195,25,26       :' JP 1A19H
```

```
10 '**********************************************************
12 '*  BASIC BLOCK TRANSFER DEMO BY CHRIS HOBROUGH - DEC '87  *
14 '**********************************************************
16 :
20 '" LOWER T.O.M. (TOP OF MEMORY) 
30 CLS:IFPEEK(30797)=128THEN100
40 TM=PEEK(30897)+256*PEEK(30898)-28
50 POKE30897,TM-256*INT(TM/256):POKE30898,INT(TM/256)
60 CLEAR 100:POKE30797,128
70 :
90 '" LOAD TRANSFER ROUTINE 
100 TM=PEEK(30897)+256*PEEK(30898):OG=TM+1
110 FORA=1TO28:READ B .
115 POKETM-65536+A,B:NEXT:AD=TM+26
120 POKETM-65536+20,AD-256*INT(AD/256)
130 POKETM-65536+21,INT(AD/256)
190 :
200 DATA 42,71,120              :'TRNSF LD   HL,(FROM)
210 DATA 237,91,73,120          :'      LD   DE,(TO)
220 DATA 237,75,75,120          :'      LD   BC,(COUNT)
230 DATA 55                     :'      SCF
240 DATA 63                     :'      CCF
250 DATA 237,82                 :'      SBC  HL,DE
260 DATA 42,71,120              :'      LD   HL,(FROM)
270 DATA 250,0,0                :'      JP   M,MVUP
280 DATA 237,176                :'      LDIR
290 DATA 24,2                   :'      JR   END
300 DATA 237,184                :'MVUP  LDDR
310 DATA 201                    :'END   RET
980 :
990 '" MAIN PROGRAM 
1000 CLS:PRINT@198,"BLOCK TRANSFER DEMO"
1010 PRINT@260,"PLEASE WAIT  25 SECONDS"
1020 :
1030 '" LOAD PATTERN INTO VIDEO RAM 
1040 PK=29184:FORX=0TO255
1042 POKEPK,128
1044 PK=PK+1:NEXTX
1050 CL=143:D=1:F=1
1052 FORI=1TO2:FORJ=1TO4
1054 FORK=1TO4:FORX=1TO8
1060 POKEPK,CL:TC=CL:CL=CL+16*D
1070 IFCL>255THENCL=143E
1075 IFCL<143THENCL=255
1080 PK=PK+1:NEXTX
1082 CL=TC:D=D*(-1)
1084 NEXTK:CL=CL+16*F
1090 IFCL>255THENCL=143
1100 NEXTJ:F=F*(-1)
1110 NEXTI:FORX=0TO255
1120 POKEPK,128
1130 PK=PK+1:NEXTX
1190 :
1200 '" FLASH PATTERN 
1210 Y=28672:Z=512:FORI=1TO10
1215 X=29184:GOSUB2000
1220 X=29440:GOSUB2000
1225 NEXTI:GOSUB3000
```

```
1490 '"USER INPUT ROUTINE
1500 CLS:PRINT@96,;:
1520 INPUT"  SOURCE ADDRESS (FROM)           ";X
1540 INPUT"  DESTINATION ADDRESS (TO)        ";Y
1560 INPUT"  NUMBER OF BYTES (COUNT)         ";Z
1580 GOSUB2000:GOSUB3000:GOSUB1500:END
1600 GOSUB2000:GOSUB3000:GOSUB1500:END
1900 :
1990 '"MOVE 'Z' BYTES FROM 'X' TO 'Y'
2000 POKE30791,X-256*INT(X/256)    'FROM -- LO BYTE
2010 POKE30792,INT(X/256)          'FROM -- HI BYTE
2020 POKE30793,Y-256*INT(Y/256)    'TO ---- LO BYTE
2030 POKE30794,INT(Y/256)          'TO ---- HI BYTE
2040 POKE30795,Z-256*INT(Z/256)    'COUNT - LO BYTE
2050 POKE30796,INT(Z/256)          'COUNT - HI BYTE
2060 POKE30862,OG-256*INT(OG/256)  'SET USR POINTERS - LO BYTE
2070 POKE30863,INT(OG/256)         'SET USR POINTERS - HI BYTE
2080 C=USR(0)                      'CALL TRNSFER ROUTINE
2090 RETURN
2990 :
3000 PRINT@483,"PRESS <RETURN> TO CONTINUE";
3010 CH$=INKEY$:CH$=INKEY$
3020 IFCH$<>CHR$(13)THEN3010ELSERETURN
```

BLOCK TRANSFERS FROM BASIC by Chris HOBROUGH. :-

Many VZ'ers who read my recent article on moving blocks of memory around may like to take advantage of the techniques but not have sufficient experience with machine language to do so. Well here is a program which will allow you to set up and call a M/L block transfer routine from basic.

The routine is the final version from the article, as it's the most versatile. It's 28 bytes long and sits in a protected area at the top of RAM. In addition, it uses 7 bytes of free space in the comms region; 3 pairs of bytes which hold the parameters for the routine (FROM, TO & COUNT) and one which is used as a flag so that memory is not lowered a bit more each time you run the program.

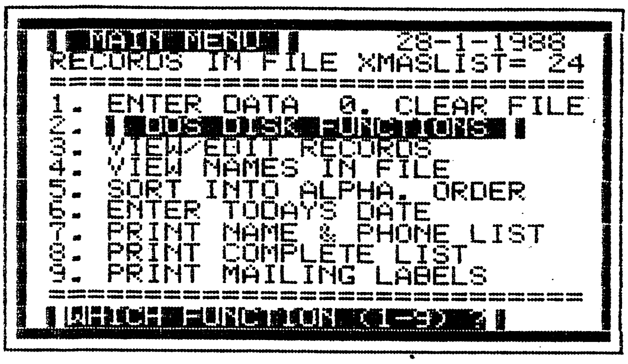
The first section lowers the Top of Memory to make room for the routine and will work for any size system. Then the routine itself is loaded into this safe area.

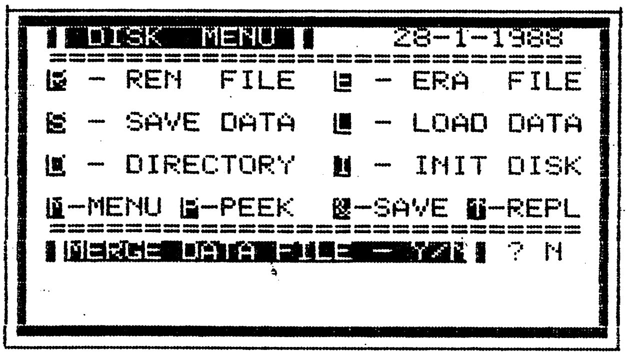
The next section, from 100 to 1600 is for demonstration only and consists of a "flashy" intro to show what can be done and a user input piece which accepts the parameters to move any block you wish. Of course you will have to put something in there to move first and also a routine to PEEK at the results.

Finally, there's the routine which passes the parameters for the move to the transfer routine and then calls it.

You can adapt this for your own needs by replacing 100 to 1600 with your own program. It must define X (FROM), Y (TO) and Z

EDITOR'S NOTE :-
This program would be ideal to make up your own HI-RES slide show. Tape users could incorporate Dave Mitchell's CSAVE/CLOAD routine from last issue.





Some of the changes made to DISKLIST are cosmetic while others are more functional. As before attempt has been made to keep to original style and the extra functions easy to use.

POKE30777,1 in lines 30, 600, 1000,7000 and 40010 are POKEs that speed up PRINTing to the screen.

POKE30777,35 in lines 645 and 1205 turn speed printing off and it must be done before disk access to stop interference with drive.

Lines 49, 2010, 2020, 3250 and 3260 add COMMAND line to ENTER DATA and VIEW NAMES screens. Lines 2020 and 3260 only need PRINT@0, changed to PRINT@32, no need to retype whole lines.

LINES 850 to 915, 5330 to 5350 and 6110 are the modified disk SAVE/LOAD routines providing two extra functions. Lines 850 to 915 are presented whole as some routines have been shifted around and line numbers have been renumbered with some lines not being altered exept for line numbers .

S - SAVE DATA - When you press 'S' an UPDATE DATA FILE Y/N option will be presented with the cursor flashing over the 'N'. Pressing RETURN will take you to normal SAVE. Entering 'Y' and pressing RETURN will allow you after entering filename to first ERAse your DATA FILE and then SAVE your DATA FILE using the same filename. In other words, automatic update.

L - LOAD DATA - When you press 'L' a MERGE DATA FILE - Y/N option will be presented with the cursor flashing over the 'N'. Pressing RETURN will take you to normal LOAD. Entering 'Y' and pressing RETURN will allow you to MERGE DATA FILES from disk with FILES in memory.

Only a few minor changes were needed to allow file merging. They are in lines 856, 868 and 6110.

0. CLEAR FILE - A clear file option has been provided. Previously this option was available only after SAVING a file, or you had to use BREAK and RUN program again. As the CLEAR FILE was already part of MAILING LIST it was easy to implement. Lines 1030, 1225, and 5330 to 5350 contain the changes.

In the Main Menu the FILENAME is now displayed and is the last change done to DISKLIST. Exept for Disk functions, most modifications can also be incorporated in MAILING LIST.

```
30 POKE30777,1:FORSC=33TO449STEP32

49 OP$="  QUIT  SCROLL  DELETE   PRINT"

600 POKE30777,1:GOSUB30:COLOR7:PRINT@34," DISK  MENU      ";DT$

645 PRINT@354," SELECT FUNCTION        ":SOUND30,1:POKE30777,35

850 PRINT@354,SP$:PRINT@380,"N"
852 PRINT@354," MERGE DATA FILE - Y/N ";:SOUND30,1:INPUTN$
854 N$=LEFT$(N$,1):IFN$<>"N"ANDN$<>"Y"THEN850
856 IFN$="N"THENOF=0ELSEOF=DT
858 IFN$="Y"THEN880ELSEIFN$="N"THEN860ELSE850
860 PRINT@354,SP$
862 PRINT@354," LOAD DATA FILE  ";:SOUND30,1:INPUTN$:GOSUB775
864 PRINT@354,SP$:PRINT@450,SP$:SOUND30,1:AD=31603:GOSUB22
866 AD=31619:GOSUB22:AD=31646:GOSUB22:AD=31674:GOSUB22
868 GOSUB6000:GOSUB20:DT=OF+DT:GOTO1000
878 :
880 IFDT<>0THEN884ELSEPRINT@353,"     NO DATA IN MEMORY!   "
882 SOUND5,7:GOTO645
884 IFDM$="L"THEN860
886 PRINT@354,SP$:PRINT@380,"N"
888 PRINT@354," UPDATE DATA FILE  Y/N ";:SOUND30,1:INPUTN$
890 N$=LEFT$(N$,1):IFN$<>"N"ANDN$<>"Y"THEN886ELSEPRINT@354,SP$
892 IFN$="N"THENUP=0ELSEUP=1
894 :
896 PRINT@354," SAVE DATA FILE  ";:SOUND30,1:INPUTN$:GOSUB775
898 PRINT@450,SP$;:SOUND30,1:IFUP=0THEN906
900 :
902 PRINT@354,SP$:PRINT@354," ERASING FILE  ";N$:SOUND30,1
904 AD=31481:GOSUB22:GOSUB12
906 PRINT@354,SP$:AD=31533:GOSUB22:AD=31549:GOSUB22
908 AD=31576:GOSUB22:AD=31674:GOSUB22:GOTO5000
915 :

1000 POKE30777,1:IFDT$=""THENDT$="NO DATE"

1020 PRINT@66,"RECORDS IN FILE          ";:PRINT@82,N$
1025 PRINT@90,USING"=###";DT
1030 PRINT@98,MI$:PRINT@130,"1. ENTER DATA  0. CLEAR FILE";

1205 POKE30777,35

1225 IFI$="0"THENIFDT=0THEN1400ELSE5260

2010 CLS:PRINTOP$
2020 COLOR8:PRINT@32,"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀";

3250 CLS:PRINTOP$
3260 COLOR8:PRINT@32,"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀";

5330 IFI$="1"THEN5350ELSEIFI$="2"THEN5340ELSE5310
5340 DT=0:ST$="":N$="":DT$="NO DATE"
5350 IFDM$="S"THEN600ELSE1000

6110 FORN=OF+1TOOF+DT

7000 POKE30777,1:GOSUB40000:GOSUB30:COLOR7

40010 POKE30777,1:CLS:COLOR8,0
```

On next page are changes to M/WORDS program to incorporate changing of pointers. Two screen dumps are shown of the MENU. First one shows changed menu while second shows pointer routine in action. Lines 725, 730, 825 and 928 to 978 are the changes to M/WORDS.

Using M/WORDS the following words were enabled :-
AUTO, CDBL, CINT, CSNG, DEFDBL, DEFINT, DEFSNG, DELETE, ERL, ERR, FIX, FRE, MEM, ON, POS, RANDOM, RESUME, STRING$, VARPTR.

No pointers had to be changed to enable above words, just inserting them in the right place did the trick. There are 4 more words that can be enabled as the routines are in ROM, but their place in the word table has been taken.

TRON-COPY, TROFF-COLOR, DEFSTR-VERIFY, ERROR-SOUND.

We can't disable existing words, but we can use some of the unused DISK WORDS from the word table. I placed the above 4 words in the the following locations in the word table. Because of compatibility I hope other VZ users adopt the following as the standard.

The tables below are in the following format :- WORD ADDRESS - WORD - TOKEN - ROUTINE ADDRESS - POINTER ADDRESS,LO-BYTE - POINTER ADDRESS,HI-BYTE. See issue # 14 for complete word table.

5856	TRON	162	7671	6246,247	6247,29
5860	ERROR	163	8180	6248,244	6249,31
5871	TROFF	166	7672	6254,248	6255,29
5905	DEFSTR	768	7680	6270,0	6271,30

The word ERROR can be used two ways. ON ERROR or ERROR X. With the changes done so far ERROR X will work, but ON ERROR will not although ON SOUND will and has same function as ON ERROR. The reason is that the word SOUND has replaced ERROR in the word table. If you type in PRINT PEEK(8045) which is second byte of ON routine the value 158 will be displayed. 158 is the TOKEN for SOUND. To get correct ON ERROR syntax is quite easy, just POKE8045,163. You'll notice 163 is the new TOKEN for ERROR.

ERROR X is used to simulate an ERROR - IE :- 10 ERROR 20
When RUN the message UNPRINTABLE ERROR will be displayed on screen.

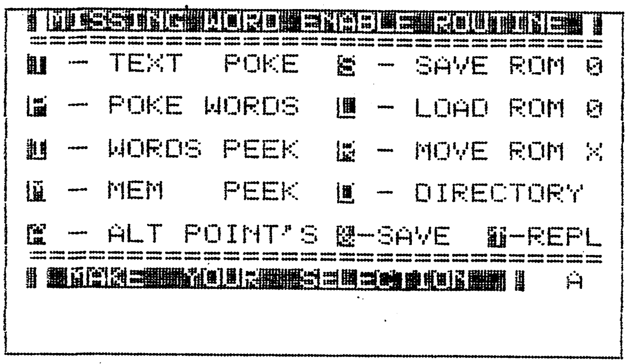
Besides the 8K BIB RAM I also have 2K BIB ram installed at 6000H-67FFH. Dave Mitchell's Extended DOS lives there as well as other routines. Three of them are Larry Taylor's modified FIND routine, Robert Quinn's PUT and GET (Block move) routines and DOS reset. Below are the words, addresses, etc.

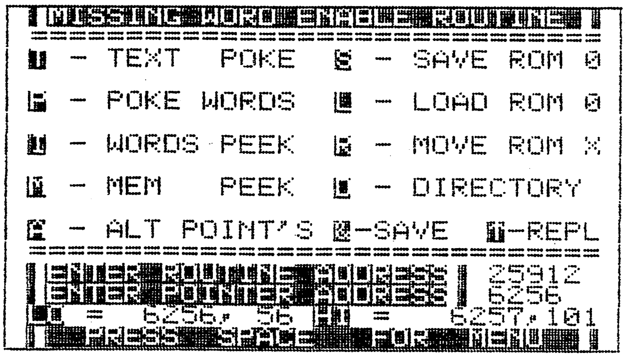
5729	DOS	133	145	6188,145	6189,0
5865	GET	164	26059	6250,203	6251,101
5868	PUT	165	26045	6252,189	6253,101
5876	FIND	167	25912	6256,56	6257,101.

DOS will work without 2K BIB RAM while others wont. Other routines can be put in 2K BIB RAM and unused disk words used like above to activate routines. On the next page is info on PUT, GET and FIND routines and how to place them in 2K BIB RAM. Your memory locations may vary, so adjust accordingly.

The BIB RAMS are ideal for developing routines and trying them out in actual operation without having to burn an EPROM first. When routines are ready, simply unplug BIB RAM and put it in your EPROM programmer/copier without losing any data because of built in batteries.

If you have any ideas on how to improve the VZ basic or DOS ROMS or have any routines that could go in 2K BIB RAM then please share with other VZ users so we can all benefit. Possible improvements - SOUND, PRINTER routines, etc.






```
725 PRINT" ■ - MEM   PEEK  ■ - DIRECTORY ":PRINT
730 PRINT" ■ - ALT POINT'S ■-SAVE  ■-REPL"

825 IFD$="T"THEN200ELSEIFD$="M"THEN200ELSEIFD$="A"THEN930ELSE800

928 :
930 PRINT@409,SP$:PRINT@385,"■ENTER ROUTINE ADDRESS■ ";:LE=5
932 SOUND30,1:GOSUB2000:IFIN$=""THEN930ELSEAD=VAL(IN$)
934 IFAD<0ORAD>32767THENPRINT@409,SP$:GOTO930
938 :
940 PRINT@417,"■ENTER POINTER ADDRESS■ ";:SOUND30,1:LE=5
942 GOSUB2000:IFIN$=""THEN940ELSEA1=VAL(IN$):A2=A1+1
944 IFA1<1ORA1>32767THENPRINT@441,SP$:GOTO940
948 :
950 PRINT@448," ■[illegible]■ ";:SOUND30,1:LE=1
952 GOSUB2000:HL$=IN$:IFHL$<>"H"ANDHL$<>"L"ANDHL$<>"Q"THEN950
954 HB=INT(AD/256):LB=AD-HB*256
956 IFHL$="Q"THEN180ELSEIFHL$="H"THENA1=A1-16384:A2=A1+1
958 GOSUB962:GOTO974
960 :
962 PRINT@449,"■ =";USING"######";A1;:PRINT",";
964 PRINTUSING"###";LB;
966 PRINT" ■ =";USING" ######";A2;:PRINT",";
968 PRINTUSING"###";HB
970 POKEA1,LB:POKEA2,HB:RETURN
972 :
974 PRINT@481,"■PRESS SPACE FOR MENU■";:GOSUB915
976 GOSUB905:GOTO180
978 :
```

```
10 '*********************************************************
20 '* FIND ROUTINE - 25912 TO 26044 - 6256,58 LO  6257,101 HI *
30 '*********************************************************
40 :
335 CLS:COLOR,0:FOR F=25912 TO 26044:READ D:POKE F,D:NEXT
340 :
345 REM - LINES 350 TO 650 FROM BASIC FIND ROUTINE
700 :
750 COLOR,1:SOUND30,1;20,3;30,1
```

```
10 '*********************************************************
20 '* PUT ROUTINE - 26045 TO 26058 - 6252,189 LO  6253,101 HI *
30 '* GET ROUTINE - 26059 TO 26072 - 6250,203 LO  6251,101 HI *
40 '*********************************************************
50 :
60 CLS:FOR P=26045 TO 26072:READ G:POKEP,G:NEXT
70 DATA 229,33,0,112,17,0,114,1,0,2,237,176,225,201
80 DATA 229,33,0,114,17,0,112,1,0,2,237,176,225,201
90 SOUND30,1;20,3;30,1:LIST-40
```

LPRINTER allows you to use your VZ as a typewriter, lprinting upper case, lower case, normal or inverse or graphic characters.

Switch on your PRINTER PLOTTER. RUN LPRINTER and a blinking cursor will appear on a blank screen to indicate your start position. Type, using any of the character keys on the keyboard by themselves or with SHIFT key held down. The corresponding character will PRINT on the screen and LPRINT to your printer.

LPRINTER starts up in normal upper case mode. Press CTRL key to shift to lower case lprinting; press CTRL again to return to upper case.

Hold SHIFT key and press X key to switch between normal and inverse printing and lprinting: inverse lprinting is distinguished from normal lprinting by underlining.

Carriage return will operate automatically to start a new line when end of line is reached, though end of LPRINT line--40 characters/line--will not correspond with end of screen line--32 characters/line. Carriage return can be accomplished anytime by pressing RETURN key.

Backspacing to start of LPRINTER line can be accomplished by holding SHIFT key and pressing B key. Every time B key is so pressed, the pen holder will move left one character. Screen cursor will backspace also.

Cursor will blink magenta color character when 35th position on LPRINTER line is reached and a warning hi-lo buzz will sound, allowing you to decide whether you can finish the current word in the remainder of that line or to execute carriage return.

Hold SHIFT and press C key then press number key (1 to 4) to select direction of lprinting (right, left, up or down):-

1 key is down
2 key is left (lprinting is upside down)
3 key is up
4 key is right

Hold SHIFT and press V key to select change of character size. Now press a number key (1 to 9). The greater the number the larger the character size selected for. If no number key is pressed then size 0 (smallest) is selected.

NOTE:- the flashing color cursor at position 35 (five characters from end of lprint line) is designed for use with a character size setting of one (40 characters /line). This is the character size LPRINTER starts up in.

Hold SHIFT key and press C key twice for SCREEN COPY, producing a printout of the entire content of the screen, normal, inverse and graphics. Character size reverts to size one for screen copy.

You can escape SCREEN COPY anytime by pressing SPACE bar/key.

With LPRINTER in memory but not RUNing, SCREEN COPY can be used by entering the command GOSUB300 and pressing RETURN key.

```
1 '*************************************************************
2 '* LPRINT UPPER/LOWER CASE, GRAPHICS AND INVERSE CHARACTERS *
3 '*  USING YOUR PRINTER/PLOTTER --- WRITTEN BY ROBERT QUINN  *
4 '*************************************************************
5 :
6 PN=PEEK(30744)
7 PK=224:POKE30862,80:POKE30863,52:IFPEEK(30744)=1THENPK=207
10 CLS:SOUND0,2:LPRINTCHR$(18);"S1":SS=-2:S6=-12:GOSUB45
15 A$=INKEY$:A$=INKEY$:IFA$<>""THENX=USR(X):GOSUB50:GOSUB45
20 IFPEEK(26875)=249THENSOUND20,1:P=NOTP
21 IFPEEK(26875)=243THENV=V-1:SOUND15,1
22 IFV=-2THENV=0:POKE30744,1:GOSUB300:POKE30744,PN
23 IFPEEK(26875)=219ANDV=1THENSOUND20,1:V=2
24 IFPEEK(26875)=219ANDV=0THENSOUND20,1:V=1
25 IFPEEK(26877)=251THENK=NOTK:SOUND20,1
26 IFPEEK(26875)=250ANDD>0THENGOSUB200:GOSUB45
27 IFC=1THENPOKEA%,T%
28 IFC=10ANDD=35THENPOKEA%,239:GOTO30
29 IFC=10THENPOKEA%,PK
30 C=C+1:IFC=20THENC=1
40 GOTO15
42 :
45 A%=PEEK(30752)+256*PEEK(30753):T%=PEEK(A%):C=10:RETURN
50 IFV=1THENV=VAL(A$):SS=-1-V:S6=SS*6:LPRINT"S";V:V=0:RETURN
52 IFV=2THEN400ELSEIFV=-1THENV=0:LPRINT"Q";VAL(A$):RETURN
55 A=ASC(A$):B=A:IFP=-1ANDA>31ANDA<64THENB=B+192
60 IFP=-1ANDA>63ANDA<128THENB=B+128
65 IFK=-1ANDA>63ANDA<95THENA=A+32
70 IFA>127THENGOSUB110:GOTO90
80 LPRINT"P";CHR$(A):IFA=13THENLPRINT"A":LPRINTCHR$(18)
90 IFP=-1ANDA<127ANDA>31THENL1=1
92 IFL1=1THENL1=0:LPRINT"R";SS*6;",0":LPRINT"P";CHR$(95)
95 IFB=13THENPRINT" ";CHR$(8);:D=-1
100 PRINTCHR$(B);:D=D+1:IFD=35ANDSS=-2THENSOUND31,2;20,1
102 IFD=INT(96/(ABS(SS)*1.2))+1THEND=1:GOSUB109
105 RETURN
107 :
109 LPRINT"A":LPRINTCHR$(18):PRINTCHR$(13);:RETURN
110 LPRINT"IR0,";-SS
120 LPRINT"J";-SS*4;",0,0,";-S6;",";SS*4;",0,0,";S6
122 S2=2*SS:S3=3*SS
125 IFA=129THENLPRINT"R";-S2;",0":GOSUB250ELSEIFA=130THENGOSUB250
130 IFA=131THENGOSUB250:GOSUB250
135 IFA=132THENLPRINT"R";-S2;",";-S3:GOSUB250
140 IFA=133THENLPRINT"R";-S2;",0":GOSUB250:LPRINT"R";S2;",";-S3
142 IFA=133THENGOSUB250
145 IFA=134THENGOSUB250:LPRINT"R0,";-S3:GOSUB250
150 IFA=135THENGOSUB250:GOSUB250:LPRINT"R";S2;",";-S3:GOSUB250
155 IFA=136ORA=137THENLPRINT"R0,";-S3:GOSUB250
160 IFA=137THENLPRINT"R0,";S3:GOSUB250
165 IFA=138ORA=139ORA=142THENGOSUB250:LPRINT"R";S2;",";-S3
168 IFA=138THENGOSUB250
170 IFA=139THENGOSUB250:LPRINT"R0,";S3:GOSUB250
175 IFA=140ORA=141THENLPRINT"R0,";-S3:GOSUB250:GOSUB250
180 IFA=141THENLPRINT"R";S2;","S3:GOSUB250
185 IFA=142THENGOSUB250:GOSUB250
190 IFA=143THENLPRINT"HP#"
195 LPRINT"HR";-SS*6;",0":RETURN
200 SOUND10,1:PRINT"";CHR$(8);:LPRINT"R";SS*6;",0":D=D-1:RETURN
250 LPRINT"J";-S2;",";-S3:LPRINT"R";S2;",0":LPRINT"J";-S2;",";S3
260 RETURN
```

```
300 A$="1":V=1:GOSUB50
305 LPRINT"A":LPRINTCHR$(18):D=0:FORT=28672TO29183:A=PEEK(T)
310 IFA<32THENLPRINT"P"CHR$(A+64)ELSEIFA<64THENLPRINT"P"CHR$(A)
320 IFA>63ANDA<96THENLPRINT"P"CHR$(A):L1=1
330 IFA>95ANDA<128THENLPRINT"P"CHR$(A-64):L1=1
335 IFL1=1THENL1=0:LPRINT"R";SS*6;",0":LPRINT"P"CHR$(95)
340 IFA>127THENGOSUB370
350 D=D+1:IFD=32THEND=0:LPRINT"A":LPRINTCHR$(18)
360 IFINKEY$=" "THENT=29184
365 NEXT:D=0:LPRINT"A":LPRINTCHR$(18):RETURN
370 IFA>143THENA=A-16:GOTO370
380 GOSUB110:RETURN
400 V=0:LPRINT"R0,";VAL(A$)*12*(-SS):PRINT" ";CHR$(8);
410 FORJ=1TOVAL(A$):PRINTCHR$(27);:NEXT:RETURN
```

M/KEYBOARD UPDATE BY JOE LEON . .

In last issue I described how to replace VZ Membrane Keyboard, also you may have read of my problems with my VZ system. Instead of of spending lot of time trying to trace problem/s I decided to upgrade my VZ 300 to No. 1 status.

Unfortunately replacing M/Keyboard did'nt cure all the problems. Some keys still required a hard (Ouch) jab to register. While installing the new PIEZO TRANSDUCER in the VZ as per article in page 5 of this issue another problem appeared with the ribbon cable connecting the P.C.B. to the keyboard.

The ribbon cables insulation parted company allowing some wires to touch each other. I had no choice but to replace the ribbon cable as it could'nt be repaired and that is when more problems arose. When the ribbon cable was replaced and before reassemling keyboard I decided to clean the P.C.B. using a pencil eraser as someone advised me that it would work.

Put everything together and only a few keys worked. Some more consultation and metho was tried with worse result. By now I was desperate as I needed a reliable VZ to prepare this issue. More consultations followed with the correct advice received at long last, thanks Dave Mitchell.

So if you're having trouble with your keyboard with some of the keys not registering then don't rush out and buy a membrane keyboard like I did. Try cleaning your P.C.B. first as it's much cheaper. The magic solution is :-

DICK SMITH SPRAY CLEANER - Cat. No. N-1501 - $3.25

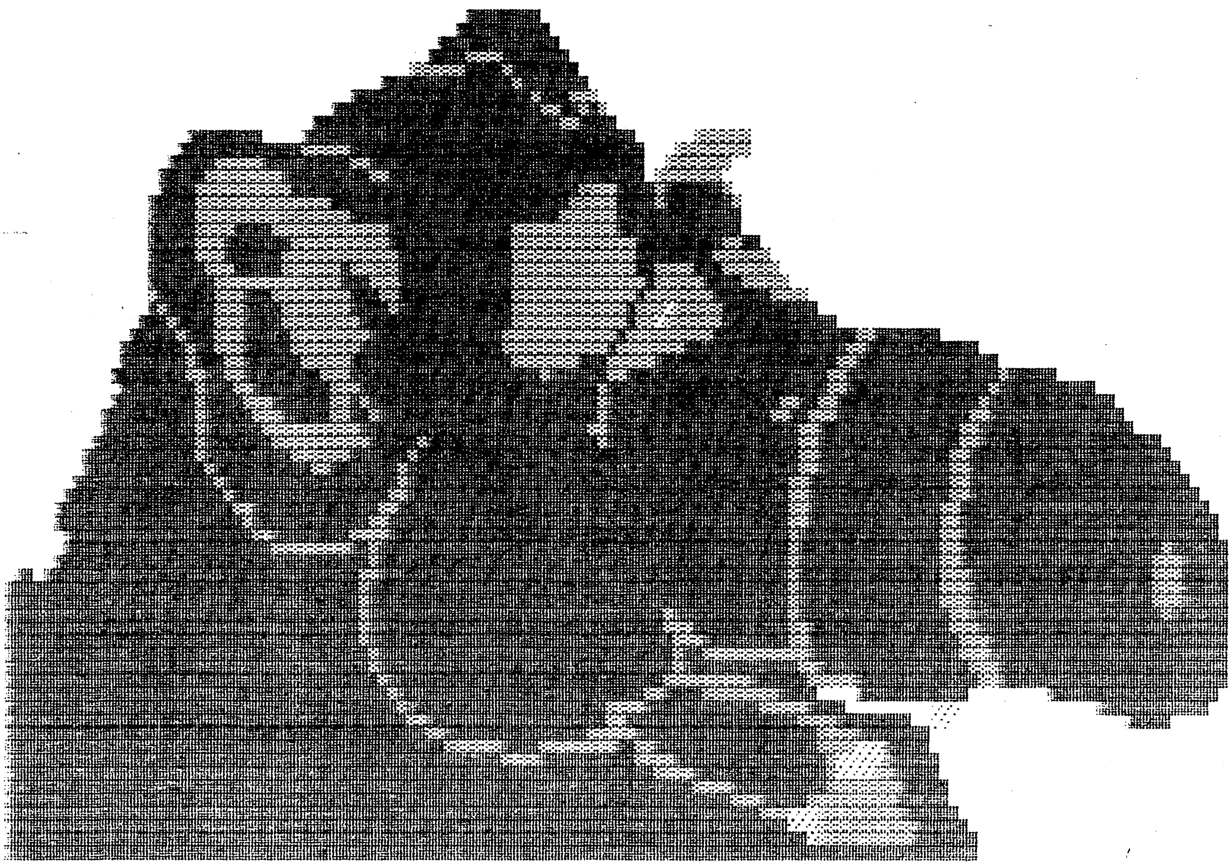
Do not spray the M/Keyboard, only the P.C.B. and wait about 5-10 minutes for the P.C.B. to dry out before reassembling. Since using the cleaner on the keyboard all the keys work and they are very sensitive to the touch now.

NOTE :- Use caution when opening up the VZ as things can and do go wrong. The two most likely problem areas are the keyboard ribbon cable and the on/off switch leads, both are brittle and break very easyly.

```
10 '*************************************************************
20 '*      FAST BLOCK MOVE & COMPARE ROUTINE BY DAVE BOYCE     *
30 '*              WITH ASSISTANCE FROM DAVE MITCHELL          *
40 '*************************************************************
90 :
100 ' STR'S YY$,YT$
110 ' VAR'S MC,L,A,B,CH,MF,EP,NS,LS,MS,P,EL,EM,NL,NM,C,G,H,D
115 :
120 '"SET UP M/C (MACHINE CODE)
130 MC=29200:P=MC:CLS
140 FOR L=1 TO 62:READ A:POKE MC,A:MC=MC+1:CH=CH+A :NEXT L
150 IF CH<> 5789 THEN SOUND 2,2:PRINT:PRINT"ERROR IN DATA":END
152 :
155 '"BLOCK MOVE DATA
160 DATA229,213,197,33,0,0,17,0,0,1,0,0,237,176,193,209,225,201
162 :
165 '"COMPARE DATA
170 DATA 0,0,229,213,197,33,0,0,17,0,0,1,0,0,26,190,32,16,35
180 DATA 19,11,120,177,32,245,62,0,50,77,114
190 DATA 193,209,225,201,62,1,50,77,114,193,209,225,201,2
195 '"THE LAST PIECE OF DATA IS A DUMMY BUT MUST BE THERE
197 :
200 CLS:GOTO300
205 :
210 '"POKE ADDRESSES INTO M/C IN VIDEO RAM
215 ' MOVE FROM   MF LS MS
220 MS=INT(MF/256):LS=MF-(256*MS)
225 '" MSB          LSB
230 POKEP+4,LS:POKEP+5,MS:POKEP+24,LS:POKEP+25,MS
232 :
235 '"NO OF BYTES TO MOVE  EP EL EM
240 EM=INT(EP/256):EL=EP-(256*EM)
245 '" MSB          LSB
250 POKEP+10,EL:POKEP+11,EM:POKEP+30,EL:POKEP+31,EM
252 :
255 '"NEW START  NS NL NM
260 NM=INT(NS/256):NL=NS-(256*NM)
265 '" MSB          LSB
270 POKEP+7,NL:POKEP+8,NM:POKEP+27,NL:POKEP+28,NM
280 RETURN:END
290 :
300 '"MOVE OR COMPARE SELECTION
310 PRINT@41,"FAST BLOCK MOVE"
320 PRINT@108,"& COMPARE"
325 PRINT:PRINT" USE ONLY POSITIVE NUMBERS EVEN  IF ABOVE 32767"
330 PRINT@260,"1) - BLOCKMOVE"
340 PRINT"    2) - COMPARE"
350 PRINT:INPUT" CHOICE ";D
360 IF D=1 THEN CLS:GOTO 400 ' MOVE & COMPARE
370 IF D=2 THEN 610          ' COMPARE ONLY
380 IF D<1 OR D>2 THEN SOUND2,2:GOTO 350
390 END
400 PRINT@41,"FAST BLOCK MOVE"
410 PRINT"   USE ONLY POSITIVE NUMBERS"
420 PRINT@134,;:INPUT"MOVE FROM ";MF
430 PRINT@193,;:INPUT"NO # OF BYTES TO MOVE ";EP
440 PRINT@257,;:INPUT"NEW BLOCK TO START AT ";NS
450 IF NS<34000 ,SOUND2,2:PRINT@281,"       ":GOTO 440
460 PRINT:INPUT" ARE THESE ENTRIES CORRECT ";YY$
470 IF YY$<>"Y" THEN RUN 'CHECK  ENTRIES
```

```
480 '"USER JUMP TO M/C BLOCK MOVE
490 GOSUB 210
500  POKE30862,16:POKE30863,114:X=USR(0)
510 PRINT:PRINT" MOVE COMPLETE"
515 :
520 '"USER COMPARE SELECT
530 PRINT:INPUT" RUN COMPARISON ROUTINE ";YT$
540 IF YT$<>"Y" THEN END ' ELSE RUN COMPARE M/C
550 POKE30862,36:POKE30863,114:X=USR(0)
560 B=PEEK(29261)
570 IF B=0THEN PRINT" ALL OK":SOUND12,1;9,1:END
580 IF B=1THEN PRINT" (NEW) BLOCK DOES NOT COMPARE"
590 PRINT" TO ORIGINAL.BLOCK":SOUND 2,4;2,1;2,4
600 END
610 CLS:PRINT@42,"FAST COMPARE"
620 PRINT:INPUT" ORIGINAL MEM AT ";MF
630 INPUT" BLOCK TO COMPARE AT ";NS
640 INPUT" COMPARE NUMBER OF BYTES ";EP
650 PRINT:INPUT" ARE THESE ENTRIES CORRECT ";YY$
660 IF YY$<>"Y" THEN RUN 610 'CHECK  ENTRIES
670 GOSUB 210:GOTO550
680 END
710 ' MAKE ALL ENTRIES IN DECIMAL
720 ' USE POSITIVE NUMBERS EVEN IF MEM ADDRESS'S ABOVE 32767
730 ' THE BLOCK MOVE ALSO ALLOWS A COMPARE,
740 ' WHERE THE COMPARE IS JUST A COMPARE
750 FOR K=29200 TO 29261:PRINTK;PEEK(K);:NEXT:END
760 :
770 ERA"FASTBM&C"    ' LINES 770 TO 790 ARE
780 SAVE"FASTBM&C"   ' FOR DISK DRIVE USERS
790 STATUS:DIR:END
```

ALF DRAWN BY MATTHEW TAYLOR . . .



EXTENDED DOS VERSION 1.0 (C) COMMANDS :-

```
MERGE - MERGES basic file from disk with program in memory.
DIRA  -  See example - T:MENU      B:PATCH3.1 B:WORDPROC
                       B:EXTDOS E B:EXTDOS R W:DOS-INST
LDIRA - As above, but to screen and printer.
DIRB  - See example  - T:MENU      01 00 7AE9 801B 0532
                       B:PATCH3.1 01 0B 7200 771F 051F
LDIRB - As above, but to screen and printer.
STATUSA  - Prints free disk space to screen on one line.
LSTATUSA - As above, but to screen and printer, see below.
           534 RECORDS FREE   63.500K FREE
OLD  - Restores a program after using the NEW command.
OLD. - Prints START, END and LENGHT of program in memory in HEX.
DEC XXXXX - Converts DECIMAL to HEX
HEX XXXX  - Converts HEX to DECIMAL
STATUSA and LSTATUSA also works with Version 1.0 DOS.
```

The EXTENDED DOS is available in the two versions below :-
EXTDOS R - T.O.M. SEEKING (SELF RELOCATING)
EXTDOS E - FOR 2K RAM AT 6000-67FF HEX
Price - $10.00 each or the two for $15.00. Availble from :-

Dave MITCHELL - (079) 27 8519
24 ELPHINSTONE STREET   NORTH ROCKHAMPTON   QUEENSLAND   4701

FOR INFORMATION IN NEWCASTLE AREA :- Joe LEON - (049) 51 2756

FOR SALE - DATABASE - DISK / TAPE

DATA - 16k - VZ DATABASE. Enter data into records thirty characters long (accepts graphic characters). Runs on VZ 200+16k or VZ 300. Available on disk as DISK DATABASE or on tape as CASSETTE DATABASE.

Facilities include data entry into record of choice, into last record chosen, next record, auto-next for fast data entry, edit keys so you don't have to re-enter entire content of a record, delete a record, delete a block of records, gap delete, insert, gap insert, fast alphabetical sort of records--start anywhere in records ; number sort ; swap any two records ; page display--ten records per page ; display current page, next page, previous page, flip backward and forward through datafile, swap any two pages, fast search of entire datafile for a sequence of characters--anywhere in records, hardcopy your records--especially suited for VZ printer plotter ; menu etc.

Disk DATA has Directory and ERASE commands, saves a datafile or any part thereof as a single binary file which loads back quickly. Cassette DATA CSAVES a datafile as a single T file--no slow loading of multitudes of D files! All instructions for using DATA are stored on disk and tape as datafiles--run DATA, load an instruction file and page through it. This program certainly stands out amongst the crowd of other such programs of it's type.

PRICE - $20.00 for DISK or CASSETTE DATABASE - Please make all Cheques and Money Orders payable to and is available from :-
SCOTT LE BRUN   5 CAMERON COURT   WANTIRNA   VIC.   3152

EXTENDED DOS VERSION 1.0 (C) - COPYRIGHT - DAVE MITCHELL - 1987

```
MERGE - MERGES basic file from disk with program in memory.
DIRA  -  See example - T:MENU      B:PATCH3.1 B:WORDPROC
                       B:EXTDOS E B:EXTDOS R W:DOS-INST
LDIRA - As above, but to screen and printer.
DIRB  - See example  - T:MENU     01 00 7AE9 801B 0532
                       B:PATCH3.1 01 0B 7200 771F 051F
LDIRB - As above, but to screen and printer.
STATUSA  - Prints free disk space to screen on one line.
LSTATUSA - As above, but to screen and printer, see below.
           534 RECORDS FREE   63.500K FREE
OLD  - Restores a program after using the NEW command.
OLD. - Prints START, END and LENGHT of program in memory in HEX.
DEC XXXXX - Converts DECIMAL to HEX
HEX XXXX  - Converts HEX to DECIMAL
STATUSA and LSTATUSA also works with Version 1.0 DOS.
```

The EXTENDED DOS is available in the two versions below :-
EXTDOS R - T.O.M. SEEKING (SELF RELOCATING)
EXTDOS E - FOR 2K RAM AT 6000-67FF HEX
Price - $10.00 each or the two for $15.00. Availble from :-

Dave MITCHELL - (079) 27 8519
24 ELPHINSTONE STREET NORTH ROCKHAMPTON QUEENSLAND 4701

FOR INFORMATION IN NEWCASTLE AREA :- Joe LEON - (049) 51 2756

FOR SALE - E & F W.P PATCH 3.1

PATCH3.1 - COPYRIGHT - HUNTER VALLEY VZ USERS' GROUP

This single Patch will convert your E & F TAPE WORD PROCESSOR for full DISK use while retaining all TAPE functions. It can be used with 1 or 2 DRIVES. Below are the two Menus.

E)DIT TEXT	L)OAD
C)LEAR TEXT	S)AVE
P)RINT TEXT	D)IR
L)OAD FILE	E)RA
S)AVE FILE	R)EN
V)ERIFY FILE	I)NIT
Q)UIT PROGRAM	1-2) DRIVE 1
D)ISK	M)ENU

Fast SAVING and LOADING of TEXT DATA to and from Disk is provided using Block SAVE or LOAD.

Full instructions are supplied together with a Tape to Disk transfer utility for your E & F Tape Word Processor.

This Patch will work with V1.0 or V1.2 Disk Controller. A STATUS facility has been added for V1.0 DOS owners.

SYSTEM REQUIREMENTS :-   DISK DRIVE + V1.0 OR V1.2 DOS
VZ300 + 16K RAM PACK OR VZ200 + 18K (16K RAM PACK + 2K)

The price - $10.00, NZ AU$12.00 and is available from :-

HUNTER VALLEY VZ USERS' GROUP
P.O.BOX 161 JESMOND 2299
N.S.W. AUSTRALIA Phone (049)51 2756